A TUTTO commodore

I.P.

Direttore responsabile Ruben Castelfranchi
Aut. alla pubblic. Trib. di Monza n. 258 del 28.11.74
STAMPA:GEMM GRAFICA s.r.l. Paderno Dugnano (MI)

GIUGNO 1985 (6)

POSTA

IL C-16 SOSTITUISCE IL VIC-20

Alla Redazione.
Negli ultimi tempi ho sentito sempre più insistentemente parlare dei nuovissimi microcomputer della Commodore, più precisamente del Commodore 16 e del Commodore Plus 4.
Parlando con i miei amici e compagni di hobby, le voci sono contrastanti e nessuno è stato capace di darmi un consiglio valido sull'acquisto di tali due apparecchi.
Dico questo perchè, ora, io sono in possesso di un Vic-20 e desidererei passare, senza spendere una montagna di quattrini, a un sistema superiore.
Vi ringrazio per la sicura risposta.

Paolo Fossati - Milano

Parlare di un microcomputer in poche righe e cercare di darne una descrizione sintetica e nello stesso tempo soddisfacente, è cosa sicuramente ardua, e si rischia quasi certamente di dimenticare di descrivere qualche aspetto importante a beneficio di altri meno significativi.
Comunque, per venire incontro al nostro lettore, cerchiamo di riassumere in poche parole ciò che Paolo ci richiede.
Il Commodore 16 e il Commodore Plus 4 sono sicuramente due computer rivoluzionari per chi possedeva un Commodore Vic-20 o un 64.
In particolare il discorso vale per il Commodore 16 che, per le sue caratteristiche di cui andremo a parlare, si presenta come il naturale sostituto del Vic-20.
Prendiamo in esame le parti che i due micro hanno in comune.
Il Basic 3.5 di cui sono dotati, si presenta arricchito di quasi 50 funzioni in più di quello del 64 e 20. Esse sono indirizzate soprattutto alla gestione della grafica, del suono, della programmazione, delle periferiche e dei joystick, e ci ricordano da vicino quelle che si possono aggiungere al 64 con il Simon's Basic.
Questa prerogativa fa sì che con semplici comandi si effettuino operazioni che, col 64 e 20, erano consentite solo attraverso delle routine, magari complicate, adatte al caso.
Non è certo un aspetto poco importante questo: infatti la poca memoria disponibile nel Commodore 16 viene così utilizzata per cose "intelligenti" risparmiandola da compiti "ingrati".
Sarebbe compito ingrato elencare tutte le nuove funzioni disponibili e, del resto,

stanno uscendo vari libri che analizzeranno entrambe le macchine in maniera più precisa e particolareggiata.
Per i più smaliziati sappiate che è in arrivo la Reference Guide per il Commodore 16 e il Commodore Plus 4.
In ogni caso Vi terremo informati sulle novità editoriali in arrivo e sicuramente ne pubblicheremo delle dettagliate recensioni.
Venendo alla domanda del nostro lettore pensiamo che, tutto sommato, uno scambio Vic-20 - Commodore 16 è sicuramente vantaggioso e da fare.
Il prezzo è sicuramente concorrenziale (Commodore 16 - 200.000 L. - Commodore Plus 4 intorno a 500.000 e nessuna macchina ha, a parità di prezzo, tali e tante prerogative.
Non va dimenticato che sono due macchine predisposte al funzionamento con il drive. Hanno infatti impostati i tasti funzione in maniera da non dover scrivere lunghi poemi per accedere al disco, e il Basic presenta numerose istruzioni rivoluzionarie per tale gestione.
In conclusione un giudizio sicuramente positivo e una previsione che vede, nel tempo, un sicuro successo per entrambi i microcomputer. Un saluto dalla Redazione.

SOFTWARE J.C.E. PER C-64

Spettabile Redazione,
sono un grande appassionato d'elettronica e solo da poco posseggo un Commodore 64. Ultimamente ho acquistato il numero di gennaio della vostra rivista "Sperimentare Computer", trovandovi con sorpresa delle offerte di programmi di carattere matematico-scientifico che mi interessano particolarmente, quali, ad esempio: "Algebra matriciale" e "Studio di funzioni" della JCE, ma disponibili solo per lo Spectrum 48K.
Vi chiedo, dunque, se programmi simili sono disponibili anche per il Commodore 64, e, se possibile, di inviarmi dei depliant in caso affermativo. Colgo quindi l'occasione per farvi i miei più sinceri complimenti per la vostra riuscita pubblicazione e spero che mi rispondiate al più presto.
Distinti saluti.

Andrea Pascuzzi
Castellammare di Stabia (Napoli)

Caro Andrea, siamo lusingati per i complimenti al nostro inserto Commodore, purtroppo per il momento i programmi di carattere matematico e didattico sono disponibili solo per lo Spectrum 48K. Sebbene il C-64 sia il più diffuso degli Home Computers gli autori di software su questa macchina non sono così numerosi come quelli dello Spectrum; ancora meno numerosi sono gli autori che hanno esperienza di C-64 e contemporaneamente conoscono sufficientemente anche lo Spectrum per poter trasferire i programmi dall'uno all'altro.

COMMODORE INGLESI

Ho ordinato un Commodore in Inghilterra, allettato da una pubblicità a prezzo interessante, trasporto e assicurazione compresi, esente da tassazione.
Naturalmente ho dovuto pagare in anticipo, e dopo aver atteso per svariate settimane, ho ricevuto un avviso che c'era per me un pacco in Dogana da ritirare.
Ho perso un sacco di tempo in quegli uffici e ho dovuto sborsare altri soldi per cui, a conti fatti, non ho risparmiato niente e mi sono preso anche delle arrabbiature.
Come spiegate questo fatto?

Giuseppe Molfetta

Si fa presto a spiegare. Quell'"esente da tassazione" riguarda l'Inghilterra, non l'Italia. La pubblicità che ti ha allettato non diceva bugie. Ma l'esenzione andava intesa nel senso che gli inglesi, esportando, non pagano sulla merce la loro VAT, corrispondente alla nostra IVA.
Quando, invece, la merce arriva alla Dogana italiana, la situazione si rovescia. Il destinatario, essendo italiano in Italia e importatore, non può ovviamente fruire dei vantaggi del mittente, che è inglese in Inghilterra ed esportatore.
La Dogana italiana è il castigamatti. Se vuoi ritirare il tuo pacco, paga l'IVA. Se no sai che cosa ti può capitare? Che il tuo pacco ritorna al mittente, perchè ufficialmente è "come se tu lo avessi rifiutato". Va a finire che gli inglesi, i quali ricevono i soldi in anticipo, se no col cavolo spediscono la merce, ricevono anche il computer di ritorno. E tu reclama, se ti conviene, perché la commedia ricomincia da capo. Dunque, o paghi l'IVA, e come hai sperimentato non risparmi nulla, ti arrabbi e perdi tempo, oppure prendi una bella fregatura e la colpa è tua.

DISALLINEAMENTO DRIVE

Spett.le Redazione,
sono un vostro superaffezionato lettore. Essendo possessore di un C-64 vi faccio i miei complimenti per l'inserto Commodore e spero che sarà ampliato.
Vi volevo fare due domande; la prima riguarda il mio drive 1541 che spesso non gradisce i dischetti dei miei amici, perché?
Poi, anche se è una domanda strana, ho sentito dire che è in preparazione un lettore laser per il mio computer. È vero? E se è vero, potete darmi qualche informazione?
Vi saluto calorosamente.

Giuseppe Fanale - Palermo

Caro Giuseppe, secondo il parere dei nostri tecnici, il tuo drive potrebbe avere un lieve disallineamento delle testine di lettura e scrittura, il che comporta il fatto che legge solo sui dischetti su cui ha scritto precedentemente, cioè i tuoi, mentre non riesce a leggere nessun segnale su quelli dei tuoi amici.
Ti consigliamo di farlo controllare da un tecnico specializzato per riallineare per bene le testine.
Per la seconda domanda purtroppo non ne sappiamo molto neppure noi: quest'estate era circolata la voce della messa a punto, in America, di un lettore laser per computer ma noi non l'abbiamo mai visto qui in Italia e dubitiamo, se esistesse veramente, che venga importato.
Ti salutiamo e ti ringraziamo per i complimenti.

TURBO TAPE PER C-64

Spettabile Redazione di Sperimentare, con la presente mi propongo di farvi alcune richieste. Gradirei che nella rubrica "A Tutto Commodore" fosse pubblicato se è possibile un eventuale metodo per accelerare il salvataggio e la lettura di file e programmi da nastro e da disco. Sono cosciente di chiedervi una cosa non semplice ma di tale interesse da giustificare la sua pubblicazione. Vi rinnovo i miei complimenti alla eccellente rivista e con la speranza che la mia richiesta sia presa in considerazione, sentitamente ringrazio.

Filiberto Foresti
S. Nicola Arcella (CS)

Caro Filiberto, siamo lusingati per i complimenti che gentilmente indirizzi alla nostra rivista e per ricambiare cercheremo di esaudire il tuo desiderio. Come ben saprai esiste un sistema di caricamento chiamato "Turbo Tape" che permette di caricare il programma molto più velocemente, circa sette volte; ogni casa produttrice di software ha il suo "Turbo" coperto da copyright. Per pubblicare quindi un programma di tipo "Turbo" dovremmo realizzarne uno completamente nuovo il che non è una delle cose più facili, ma possiamo rassicurarti che il tuo consiglio non cadrà nel vuoto e in uno dei prossimi numeri avrai una gradita sorpresa con la pubblicazione di un "Turbo" per cassetta e uno per il disco. Meglio di così non potevamo certo fare, ci congediamo invitandoti a scriverci ancora per i tuoi prossimi eventuali problemi.

SOFTWARE HES

PER COMMODORE VIC 20

COCO 2
Un fantastico computer games di tipo educativo, che ti permetterà di crescere da solo i tuoi giochi. Coco 2 ti insegna la teoria dei giochi, la grafica dei computers, i simboli della programmazione e il disegno logico. Coco 2 non richiede alcuna precedente esperienza di programmazione.
Supporto: cassetta e disk
Configurazione richiesta: 16K RAM, registratore o floppy disk 1541
TF/9316-03 — **L. 12.500**

PREDATOR
Fiero come un'aquila, voli nel cielo, difendendo il tuo spazio dagli attacchi di nemici pennuti. Mano a mano che sconfiggi i tuoi avversari, avanzi lungo la scala di 99 livelli di difficoltà. Come un'aquila aspiri a raggiungere altezze sempre più grandi. Un gioco per tutte le età con concetti inediti e differenti dai tradizionali.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
TF/9316-40 — **L. 12.500**

TORG
TORG è un'avventura all'interno di un intricato labirinto. Intrappolato dentro una griglia devi assolutamente cercare la via d'uscita distruggendo le varie ondate di mostri che ti attaccano per annientarti.
Supporto: cassetta
Configurazione richiesta: RAM standard, joystick
TF/9316-01 — **L. 3.900**

RAID ON ISRAM
Una missione di ricerca e distruzione controllata interamente dal joystick. Con l'aiuto dei lanciarazzi, piloti il tuo caccia in una fantastica avventura ai limiti del reale.
Supporto: cassetta
Configurazione richiesta: RAM standard, joystick
TF/9316-02 — **L. 3.900**

PROTECTOR
Le orde della città di Fraxullan stanno attaccando la tua città e catturando gli abitanti.
Tu devi aiutare la popolazione, trasportandola nella Città della Nuova Speranza, sull'altro lato del vulcano; ma i sopravvissuti non sono ancora salvi.
Ogni persona dovrà essere nuovamente trasportata nella fortezza di Verdann prima che il vulcano esploda, travolgendoli tutti.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
TF/9316-36 — **L. 12.500**

AGGRESSOR
Come un valoroso pilota devi proteggere i depositi di benzina di Stellarium dagli attacchi delle navicelle di Zaurien.
Stai attento alle mine spaziali e ai vari tipi di astronavi nemiche.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
TF/9316-28 — **L. 12.500**

PER COMMODORE 64

MINNESOTA FATS' POOL CHALLENGE
Un simulatore di biliardo con tutte le caratteristiche reali di un vero biliardo. Il gioco richiede tutta la precisione, la mira e gli effetti come un biliardo da sala giochi.
Sono previsti 1 o 2 giocatori con 6 varianti di gioco.
Supporto: cartridge
Configurazione richiesta: standard, joystick
TF/9200-13 — **L. 26.500**

COCO
Eccezionale computer games di tipo educativo, che insegna le tecniche fondamentali del computer, il linguaggio basic, come risolvere i problemi scomponendo ogni programma in semplici pezzi e simulando l'effetto di ogni istruzione.
Coco ti permette di capire come lavora un computer.
Supporto: cassetta
Configurazione richiesta: standard
Istruzioni in italiano.
TF/9205-02 — **L. 12.500**

BENJI'S SPACE RESCUE
Assieme al cane Benji' stella di molti sceneggiati televisivi americani, intraprenderà un viaggio attorno al sistema solare, alla ricerca di alcuni scienziati rapiti. Scoprirai le meraviglie del cosmo, e combatterai contro alieni, insieme al fedele Benji. Dovrai mettercela veramente tutta per scoprire dove sono tenuti prigionieri gli scienziati. Ce la farai?
Supporto: diskette
Configurazione richiesta: floppy disk 1541
TF/9205-06 — **L. 7.000**

THE PIT
In questo famoso gioco arcade bisogna condurre l'avventuriero nel tunnel sotterraneo di un pianeta alieno alla ricerca del favoloso forziere che però contiene PIT, una trappola mortale dalla quale nessuno è ancora riuscito a sfuggire. Un feroce e crudele robot ti renderà la vita impossibile inseguendoti per tutto il percorso.
Supporto: cartridge
Configurazione richiesta: standard, joystick
TF/9200-16 — **L. 26.500**

RETROBALL
L'emozionante e divertente gioco dell'Hockey su ghiaccio riproposto ora sul computer.
Possibilità di giocare fra 2 giocatori o contro il computer.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
TF/9200-02 — **L. 7.500**

GHOST MANOR/SPIKE'S PEAK
Due splendide avventure grafiche al prezzo di una.
In Ghost manor il giocatore deve affrontare spettri e scheletri, mostri mummie e formiche mortali per salvare l'amico dal Conte Dracula. In Spike's Peak una straordinaria sequenza di catastrofi naturali e pericoli mortali si susseguono e impediscono il giocatore nella conquista della cima della montagna.
Supporto: diskette
Configurazione richiesta: floppy disk 1541
TF/9200-11 — **L. 22.000**

ULTERIORE SCONTO DEL 10% A CHI ACQUISTERA' 3 GIOCHI

Cod. progr.	Q.tà	Cod. progr.	Q.tà	Cod. progr.	Q.tà

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome
Cognome
Via
Città
Data — C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA
Partita I.V.A.

PAGAMENTO:
A) Anticipato, mediante assegno bancario per l'importo totale dell'ordinazione.
B) Contro assegno, in questo caso, è indispensabile versare un acconto di almeno il 50% dell'importo totale mediante assegno bancario. Il saldo sarà regolato contro assegno.
AGGIUNGERE: L. 5.000 per contributo fisso.
I prezzi sono comprensivi di I.V.A.

DIVIS. **EXELCO** Via G. Verdi, 23/25 — 20095 – CUSANO MILANINO – MILANO

SPEECH 64
Sintetizzatore vocale per Commodore 64

Grazie alle potenti capacità sonore del Commodore 64 si è riusciti a dar la parola a questo affascinante personal.

Molti saranno abbastanza scettici sul fatto che un computer di queste dimensioni possa realmente generare delle parole dette in modo più che comprensibile. È questa era anche la nostra tendenza prima di aver provato il sintetizzatore vocale "Speech 64" della ditta britannica Currah Computor Components e disponibile in Italia presso tutti i negozi GBC a lire 109.000 + IVA. Ci siamo infatti ricreduti dopo aver sentito realmente una serie di parole dette dal computer in maniera eccellente. Il sintetizzatore vocale va inserito nella porta riservata dal Commodore 64 alle espansioni. L'operazione si conclude con il collegamento del cavetto posto nella parte posteriore dell'interfaccia alla presa audio-video. In queste condizioni si può accendere il computer ed eseguire una semplice istruzione per sentire il Commodore 64 "parlare". Inserendo nel computer il comando INIT lo si abiliterà a ripetere il suono prodotto da qualsiasi tasto della tastiera. Premendo RETURN, il calcolatore ripeterà la parola con un perfetto accento americano. È a questo punto che cominciano ad esserci le prime grane. Fino a quando si fa ripetere al computer parole in inglese non ci sono problemi, ma il difficile è far ripetere parole italiane dato che l'accento resta sempre americano soprattutto per alcune lettere particolari come "R" oppure "T". Ma senza andare a prendere parole estere per SPEECH 64 (ricordiamoci che è un prodotto inglese!), ci accorgiamo che sbaglia anche le più semplici leggi di fonetica. Se infatte provate a scrivere "Kind", il computer vi risponderà ignorantemente con un secco "Kind".

Logicamente l'autore ha pensato di produrre un sintetizzatore vocale e non un maestro di grammatica. E giustamente, non essendo possibile inculcare nell'interfaccia una EPROM con migliaia di regole grammaticali, si è pensato di produrre un linguaggio fonetico in modo tale da far leggere al

computer particolari parole. Se per esempio vogliamo sentire la parola "Hawaii"dobbiamo inserire nel computer "hawhy'ee". Può sembrare un metodo un po' macchinoso per manipolare le parole, ma è anche l'unico sistema di far parlare il computer anche in altre lingue. Un'altro elemento molto interessante è la possibilità di far dire al computer parole con due tonalità di voce diverse, una alta e una bassa (non tutte le parole vengono fedelmente riprodotte con una sola voce).

Per coloro che hanno intenzione di comprare SPEECH 64 gli consigliamo di cominciare a far dire al computer prima di tutto semplici numeri o lettere e poi passare alle parole per finire alle frasi intere di senso compiuto. La lettera della tastiera viene abilitata con il comando KON e disabilitata con KOFF. Le impressioni che ci ha suscitato questa interessante interfaccia sono molto positive, considerando anche le possibili applicazioni di questa periferica. Un po' robotico, ma tutto sommato il 64 è stato arricchito da con un ottima voce.

NUOVA STAMPANTE COMMODORE MPS 802

La Commodore ha recentemente presentato una nuova stampante, che si affianca a quelle già disponibili, come la MPS-801 o la printer plotter. Questa macchina presenta delle particolarità, che la rendono estremamente interessante.

Fisicamente; la stampante si presenta come un parallelepipedo realizzato in materiale plastico dello stesso colore del C-64.

Sulla destra c'è il tasto per l'accensione, mentre sul frontale, accanto alla targhetta, è presente il tasto per l'avanzamento automatico della carta.

La stampante è ovviamente ad impatto, a matrice di punti.

La matrice è di 8x8 punti, quindi più grande delle matrici normalmente usate. Ul numero di colonne di stampa è come al solito 80. La velocità di stampa varia a seconda del numero di colonne di stampa visualizzate, da un minimo di 45 linee al minuto per 80 colonne, a 124 linee al minuto se si usano solo 20 colonne di stampa. La stampa è bidirezionale e di ottima fattura.

Il set di caratteri comprende le lettere maiuscole, minuscole (con i discendenti), i simboli grafici del Commodore e altresì i caratteri definiti dall'utente.

Oltre a ciò la macchina può effettuare stampa grafica.

La distanza fra le linee è programmabile.

Insieme alla macchina, viene fornito un manuale di una quarantina di pagine, che spiega nel dettaglio le varie funzioni e possibilità della stampante, che in alcuni casi superano le

possibilità dell'unità centrale alla quale è collegata.

I comandi per il funzionamento della stampante sono quelli soliti, con alcune particolarità però.

La OPEN, che apre il flusso di comunicazione con la stampante ha tre parametri. Il primo indica il numero del file, il secondo il numero della periferica, mentre il terzo indica il formato di stampa scelto.

Questo terzo numero, che può variare fra 0 e 10, ha diversi significati. Lo 0 indica alla stampante di effettuare la stampa dei caratteri, nello stesso modo in cui li riceve in maiuscolo/grafico, mentre il 7, fa la stessa cosa con i caratteri maiuscoli/minuscolo. Altri numeri definiscono il numero di linee per pagina, la distanza fra queste linee.

Come abbiamo precisato all'inizio, la macchina ha però una particolarità, e cioè la possibilità di utilizzare una funzione di PRINT USING (anche se solo su stampante), permettendo così all'utente di definire il proprio formato di stampa.

Ovviamente questa possibilità permette di risolvere molti dei problemi relativi ad esempio alla stampa di fatture commerciali, e altri documenti di questo tipo.

La formattazione dei dati, avviene aprendo un secondo canale di comunicazione, che serve a specificare il formato dei dati che verranno stampati attraverso il normale canale di comunicazione.

I canali aperti contemporaneamente per dare le diverse direttive possono essere più di uno, come abbiamo già visto.

Oltre a quello di trasmissione dati e trasmissione formato di stampa, se ne possono aprire altri, con il compito ad esempio di trasmettere il numero di linee per pagina da utilizzare.

In genere questi comandi vengono utilizzati contemporaneamente a dei codici di comando della stampante.

Risulta quindi abbastanza complesso, perlomeno inizialmente, riuscire a sfruttare pienamente le caratteristiche della MPS 802, proprio a causa delle numerose possibilità offerte dalla stessa e dal modo un po' macchinoso che viene utilizzato per impartire i comandi. Le difficoltà si hanno inoltre a causa del funzionamento solo su stampante di questi comandi, per cui all'utente non è dato modo di visualizzare prima sul video, il risultato delle operazioni che sta eseguendo.

Inutile comunque sottolineare il fatto, che vale proprio la pena di compiere lo sforzo di apprendere le modalità operative di questa stampante, per potere beneficiare di tutte le possibilità da essa offerte, e tali da trasformare il modo stesso di operare del Commodore, e quindi anche il modo di fare programmi.

La tabella dei caratteri di controllo è presente anch'essa sul manuale, sul quale sono altresì riportati numerosi esempi esplicativi delle modalità operative necessarie per combinare fra loro questi comandi.

La macchina può stabilire anche caratteri ridefiniti, che possono essere realizzati come normalmente avviene sullo Spectrum, utilizzando una matrice di 8x8 punti, nella quale si vanno a calcolare i valori decimali dei punti settati o meno.

La macchina può altresì stampare in reverse, o con dei caratteri con doppia larghezza.

Entrambe queste prestazioni si ottengono o con codici di controllo o utilizzando i comandi tipici del video.

★★

"BREAK FEVER" IL NUOVO PROGRAMMA DELLA INTERCEPTOR SOFTWARE

La Interceptor Software ha realizzato per Commodore 64 un fantastico programma di animazione e musica con tema la modernissima e trascinante Break Dance. Il programma è dotato di un'ottima grafica e di una colonna sonora di sottofondo che ha pochi uguali, il programma sarà disponibile su disco e cassetta, in Italia non è ancora annunciata la disponibilità e il prezzo.

NEI MEANDRI DEL COMMODORE 64

a cura di Franco Francia

GLI SPRITES: SECONDA PARTE

Riprendiamo il nostro viaggio immaginario all'interno del COMMODORE 64 continuando a parlare di quei fantastici blocchetti definibili e semovibili meglio conosciuti sotto il nome di SPRITES (folletti). La volta scorsa abbiamo illustrato quelle che sono le loro caratteristiche di base, ovvero:
(*) indipendenza totale dalla memoria video e da qualsiasi scenario in alta risoluzione presente sullo sfondo;
(*) facilmente definibili dall'utente; la tediosità delle operazioni di calcolo dei dati che li compongono può essere scavalcata per mezzo di appositi programmi ausiliari denominati 'SPRITE EDITORS';
(*) possibilità di gestirne il movimento e la relativa velocità di spostamento;
(*) permettono un considerevole risparmio di memoria rispetto all'impiego di caratteri ridefiniti;
(*) possibilità di diagnosticare qualsiasi loro collisione, sia con lo sfondo che con altri sprites; questa caratteristica è alla base delle evoluzioni di un videogame;
(*) definibili in uno dei sedici colori disponibili (monocolori), oppure in tre colori a scelta, ma con risoluzione orizzontale dimezzata (multicolori). Dopo aver visto come sviluppare il calcolo dei dati che li compongono, inizieremo questo mese ad occuparci della loro implementazione all'interno di un programma, ovvero come gestirli in seguito alla definizione della struttura grafica vera e propria. In conclusione, come promesso, vi presenteremo il listato di uno sprite editor multicolore.

PROGRAMMAZIONE DI UNO SPRITE PASSO DOPO PASSO

La prima tappa del nostro programma, come già illustrato precedentemente, consiste nella definizione dei dati che formeranno il nostro sprite. Questi valori numerici, 63 in totale, vengono gestiti sotto forma d'istruzioni DATA, in modo che il computer le possa leggere per mezzo di un comando READ e quindi immagazzinarle in memoria. Una piccola e buffa creatura aliena verrà visualizzata per mezzo dei seguenti valori:

```
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 126, 0,
     0, 255, 0, 1, 255, 128
DATA 3, 255, 192, 3, 255, 192, 3, 213, 192,
     3, 255, 192, 3, 255, 192
DATA 1, 255, 128, 0, 255, 0, 0, 66, 0, 0, 66, 0,
     0, 231, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0
```

Queste istruzioni DATA possono essere inserite in qualsiasi punto del nostro programma, ne più ne meno come ogni altra forma di dati numerici da memorizzare; è tuttavia consigliabile posizionarle in coda ad esso, in modo da evitare a priori possibili fonti di interferenza o addirittura autodistruzioni accidentali.

La successiva tappa del nostro lavoro equivale alla ricerca di una zona appropriata della memoria all'interno della quale porre i precedenti dati, segnalando contemporaneamente al computer dove trovarli per mezzo di un assegnamento ad un apposito puntatore.

Il VIC II chip, ovvero quello che si occupa della gestione del video, è in grado di accedere soltanto a 16 Kbytes per volta, all'interno dei quali devono trovarsi contemporaneamente la memoria di schermo, il generatore di caratteri e l'informazione riguardante gli sprites. In via puramente teorica, in un tale banco potrebbero essere contenuti 128 sprites; in realtà, non tutte le 128 posizioni sono utilizzabili, in quanto memoria di schermo e generatore di caratteri limitano sensibilmente la memoria disponibile. In funzione del banco di 16 Kbytes prescelto, altre locazioni possono risultare inaccessibili.

All'accensione della macchina, il banco assegnato è quello delimitato dalle locazioni 0 e 16383. In esso, la maggior parte dei primi 1024 bytes è riservata a funzioni interne o di controllo; la memoria di schermo è posta da 1024 a 2023; il generatore di caratteri, benché residente in ROM, ha un'immagine da 4096 a 8191; infine, non bisogna dimenticare che a partire da 2048 si troverà il nostro programma BASIC. Come potete ben vedere, è necessario prestare molta attenzione nel scegliere dove locare i dati degli sprites; nel caso vogliate difinirne un numero considerevole, vi consigliamo di spostare più avanti l'inizio del BASIC, oppure di passare ad un altro banco di 16 Kbytes che non interferisca con le vostre operazioni. Per il momento, ci limiteremo comunque al solo impiego del banco zero iniziale, in quanto più semplice da trattare. Dopo queste doverose premesse, vediamo adesso dove far memorizzare dal programma i dati degli sprites che utilizzeremo; nel caso questi non oltrepassino il numero di tre, un ottimo posto è rappresentato dal buffer di cassetta, situato da 832 a 1023, a condizione di non usare quest'ultima periferica per trasferimenti d'informazione. Uno sprite si piazzerà quindi da 832 a 894, un secondo da 896 a 958, ed un terzo da 960 a 1022. In caso si desideri utilizzare il registratore, oppure si disponga di più di tre sprites, è necessario cambiare zona di memoria; un indirizzo consigliabile è 12288, in quanto è normalmente abbastanza elevato da non interferire con il programma BASIC. Per essere sicuri di non interferire, è possibile proteggere quest'area per mezzo di:
POKE 52,48:POKE 56,48

In questo modo, il BASIC (tanto il programma quanto le variabili) non potrà mai oltrepassare tale indirizzo di memoria, salvaguardando così l'integrità dei nostri dati.

Dopo aver deciso dove sistemare gli sprites da utilizzare, dobbiamo comunicare al computer dove andare a prelevare i valori interessati. A questo ci pensano 8 particolari locazioni (una per ogni sprite indirizzabile) denominate 'puntatori di sprite', e situate in fondo alla memoria di schermo. Tali locazioni, inizialmente poste da 2040 (sprite zero) a 2047 (sprite sette), dipendono dal posizionamento di quest'ultima, motivo per cui non sono assolute. Come ben sapete, un byte può contenere valori compresi fra zero e 255; il numero di blocchi di 64 bytes contenuti in un banco di 16 Kbytes (quello gestito dal VIC II chip) è esattamente 256 (16384/64 = 256). Il valore immagazzinato nel relativo puntatore indicherà pertanto al computer il numero del blocco di 64 bytes contenente i dati dello sprite considerato. Vediamo un piccolo esempio:
POKE 2040,192
indicherà al computer che i dati riferiti allo sprite zero sono situati nel centonovantaduesimo blocco di 64 bytes, ovvero a partire dall'indirizzo (192*64 = 12288).

Vi rammentiamo che soltanto i primi 63 bytes del blocco caratterizzano lo sprite in questione, mentre il sessantaquattresimo non viene utilizzato.

Generalmente, i puntatori di sprite corrispondono alle ultime otto locazioni del Kbyte comprendente la memoria di schermo, spostando la quale anche questi ultimi vengono alterati.

Un esempio dell'estrema utilità di questi puntatori è dato dal fatto che, per creare una forma di animazione, è sufficiente ri-

```
10 REM ******************************
12 REM **** SPRITE EDITOR MULTI****
14 REM ******************************
15 REM
16 REM
17 REM ---------------------------
18 REM    COPYRIGHT (C) 1985 JCE
19 REM ---------------------------
20 REM
21 REM
22 REM
25 REM LA PRIMA VOLTA PROVA A SELEZ
   IONARE(IN ORDINE), I COLORI 1, 9
    E 12
26 REM PER DARE UN'OCCHIATA ALLA CO
   MBINAZIONE FAVORITA DELL'AUTORE.
29 REM
30 REM
31 REM
40 POKE828,0
45 V=53248:NO=13:SP=13:POKEV+21,0
50 READFF
60 IFFF>0THEN804
69 REM -------------------
70 REM NON PIU' SPRITE DATA
71 REM -------------------
80 POKE53281,0:POKE53280,0:PRINT"■"
85 GOSUB3000
90 DEFFNA(ZZ)=1065+R*40+C
110 XL=0:YL=1:XG=16:SE=21:XY=23:XX=2
   9
120 SC=39:PRINT"■"
130 POKE2040,13:POKEV+SE,1:POKEV+YL,
   0
140 POKEV+XX,1:POKEV+XL,255:POKEV+YL
   ,190
150 POKEV+XG,0
160 X=255:Y=190
165 POKEV+28,1
169 REM -------------------
170 REM SCHERMO DI CODIFICA
171 REM -------------------
180 PRINT"■■ ______________________
   _":REM CBM P
181 FORI=0TO20
182 PRINT"                         ":
   REM CBM N E CBM H
183 NEXTI:PRINT" ‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
   ‾‾‾‾‾‾‾":REM CBM Y
184 GOSUB1000:GOTO309
309 REM ---------------------------
   ---
310 REM SPRITE CODIFICATO SULLO SCHE
   RMO
320 REM CAMBI INPUT
321 REM ---------------------------
   ---
330 R=0:C=0
340 Z=FNA(0)
342 PC=PEEK(Z):PD=PEEK(Z+1):PK=Z
343 IFPC=160THENRV=128:GOTO345
344 RV=0
345 POKEZ,24+RV:POKEZ+1,24+RV
346 KN=PEEK(Z+54272):KM=PEEK(Z+54273
   ):IFKN=0THENKN=1:KM=1
350 POKEZ+54273,KN:POKEZ+54274,KM
360 GETA$:IFA$=""THEN360
370 POKEZ+54273,KN:POKEZ+54274,KM:PO
   KEPK,PC:POKEPK+1,PD
380 IFA$="F"THENPRINT"■■SPERO CHE TU
    ABBIA MESSO IL TUO SPRITE  NEI
   DATA!":END
390 IFA$="■"ANDC=22THENC=0:GOTO340
400 IFA$="■"THENC=C+2:GOTO340
410 IFA$="■"ANDC=0THENC=22:GOTO340
420 IFA$="■"THENC=C-2:GOTO340
430 IFA$="■"ANDR=20THENR=0:GOTO340
440 IFA$="■"THENR=R+1:GOTO340
450 IFA$="□"ANDR=0THENR=20:GOTO340
460 IFA$="□"THENR=R-1:GOTO340
470 IFA$="■"THENR=0:C=0:GOTO340
480 IFA$="■"THENGOSUB1150:GOTO340
490 IFA$="+"THEN580
500 IFA$="-"THEN730
510 IFA$="S"THEN1210
520 IFA$="B"THEN1450
530 IFA$="C"THEN1400
540 IFA$="E"THEN900
570 GOTO340
574 REM -------------------
575 REM AGGIUNGE UN PUNTO
576 REM -------------------
580 Z=FNA(0)
590 IFCS=A2THENK1=224:K2=160
595 IFCS=A4THENK1=160:K2=224
600 IFCS=A3THENK1=160:K2=160
610 POKEZ,K1:POKEZ+1,K2:POKE55296+Z-
   1024,CS:POKE55296+Z-1023,CS
620 BYTE=INT(C/8)+R*3:GOSUB625:GOTO3
   40
625 CH=1065+40*R+8*INT(C/8):TC=0:BI=
   8
626 FORL1=CHTOCH+7:P=PEEK(L1):BI=BI-
   1:IFP=160ORP=24THENTC=TC+2↑BI
627 NEXTL1:POKEBYTE+NO*64,TC:RETURN
724 REM -------------------
725 REM CANCELLA UN PUNTO
726 REM -------------------
730 Z=FNA(0)
740 Z1=PEEK(Z)
760 POKEZ,32:POKEZ+1,32
770 BYTE=INT(C/8)+R*3:GOSUB625:GOTO3
   40
804 REM ---------------------------
   ---
805 REM SE CI SONO DATA CODIFICA SPR
   ITE
806 REM ---------------------------
   ---
808 READA1,A2,A3:IFA1=0THENPOKEV+28,
   PEEK(V+28)AND255-2↑(0):GOTO810
809 POKEV+28,PEEK(V+28)OR2↑(0):POKEV
   +37,A2:POKEV+38,A3
810 READA4:POKEV+39,A4
815 LOC=SP*64
820 FORI=LOCTOLOC+62
830 READA:POKEI,A
840 NEXTI
850 GOTO50
894 REM -------------------
895 REM INPUT PER ESPANSIONE
896 REM -------------------
900 PRINT"■■■■■■■■■■■■■■■■■■■■■■■
   ■■■■■■■BATTI X O Y"
```

```
910 GETA$:IFA$<>"X"ANDA$<>"Y"THEN900
920 IFA$="X"THEN960
930 IFPEEK(V+XY)=1THENPOKEV+XY,0:GOT
    O980
940 POKEV+XY,1
950 GOTO980
960 IFPEEK(V+XX)=1THENPOKEV+XX,0:GOT
    O980
970 POKEV+XX,1
980 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    ■■■■■■■■■           "
990 GOTO340
994 REM ----------------------------
995 REM MOSTRA OPZIONI PER IL CONTRO
    LLO
996 REM ----------------------------
    ---
1000 PRINT"■■"SPC(27)"■CONTROLLI■"
1005 PRINTSPC(26)"SPRITE NO. 0"
1030 PRINTSPC(26)"■S■POSTA SPRITE"
1040 PRINTSPC(26)"■C■AMBIA COLORE"
1050 PRINTSPC(26)"■E■SPANDE"
1060 PRINTSPC(26)"■+■AGGIUNGE PNT"
1070 PRINTSPC(26)"■-■CANCELLA PNT"
1080 PRINTSPC(26)"■B■ASIC DATA"
1090 PRINTSPC(26)"■F■INE"
1100 PRINT:PRINTSPC(26)"USA LE FRECCE
     "
1110 PRINTSPC(26)"PER CAMBIARE"
1120 PRINTSPC(26)"LA POSIZIONE"
1130 PRINTSPC(26)"DEL CURSORE."
1140 RETURN
1144 REM ---------------------------
1145 REM CANCELLA IL PRESENTE SPRITE
1146 REM ---------------------------
1150 FORI=0TO62:POKENO*64+I,0:NEXTI
1160 FORI=0TO20
1170 FORJ=0TO24
1180 POKE1064+I*40+J,32
1190 NEXTJ:NEXTI:R=0:C=0
1200 RETURN
1204 REM ---------------------------
     --
1205 REM MUOVE LO SPRITE PER LO SCHER
     MO
1206 REM ---------------------------
     --
1210 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■CU
     RSORE MUOVE SPRITE/RETURN PER US
     CIRE."
1230 GETA$:IFA$=""THEN1230
1240 IFA$="■"ANDX<319THENX=X+2
1250 IFA$="■"ANDX>1THENX=X-2
1260 IFA$="■"ANDY<254THENY=Y+2
1270 IFA$="■"ANDY>1THENY=Y-2
1280 POKEV+YL,Y
1290 POKEV+XG,INT(X/255)
1300 POKEV+XL,X-INT(X/255)*255
1310 IFA$=CHR$(13)THEN1330
1320 GOTO1210
1330 POKEV+XL,255
1340 POKEV+YL,190
1350 POKEV+XG,0
1360 X=255:Y=190
1370 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
     "
1390 GOTO340
1394 REM ---------------------------
1395 REM CAMBIA COLORE ALLO SPRITE
1396 REM ---------------------------
1400 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■F
     AI LA TUA SCELTA (1, 2 O 3)."
1402 GETCG$:IFCG$=""THEN1402
1403 IFCG$="1"THENCS=A2:GOTO1420
1404 IFCG$="2"THENCS=A3:GOTO1420
1405 IFCG$="3"THENCS=A4:GOTO1420
1406 GOTO1402
1420 POKEV+36+VAL(CG$),CS
1430 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
     ■";
1440 GOTO340
1444 REM ----------------------------
     -
1445 REM CREAZIONE ISTRUZIONI DATA PE
     R
1446 REM LO SPRITE PRESENTE
1447 REM ----------------------------
     -
1450 PP=PP+11:PRINT"■■■■";PP+30000;"D
     ATA";SP;",1,";A2;",";A3;",";A4;"
     "
1460 FORI=0TO8
1470 PRINTPP+I+30001"DATA";
1480 FORJ=0TO6
1490 BB=PEEK(NO*64+I*7+J)
1500 BB$=RIGHT$(STR$(BB),LEN(STR$(BB)
     )-1)
1510 PRINTBB$;",";
1520 NEXTJ
1530 PRINT"■ "
1540 NEXTI
1550 PRINTPP+10+30000;"DATA -1":NO=NO
     +1:SP=SP+1
1560 PRINT"GOTO045■"
1570 POKE198,12
1580 FORI=0TO11:POKE631+I,13:NEXTI
```

assegnarlo ad un nuovo blocco contenente i dati del successivo disegno. Si continuerà in tal modo ad utilizzare sempre il medesimo sprite, il quale cambierà immediatamente fisionomia per mezzo di una semplice istruzione POKE. Inutile sottolineare come la rapidità di esecuzione sia alla base di qualsiasi forma di grafica animata; pensate soltanto per un istante ai rallentamenti provocati dall'eventuale riassegnamento di ben 63 bytes, con evidenti disagi provocati tanto alla struttura del programma quanto al risultato estetico finale.

SPRITE EDITOR MULTICOLORE

Questo listato conserva lo stesso formato del precedente sprite editor monocolore, comprendendo tuttavia sostanziali modifiche relative agli sprites multicolori.

Le linee sulle quali occorre puntare la maggior attenzione sono quelle comprese fra 625 e 627, nelle quali vengono calcolati i dati dei vari sprites in assemblaggio.

Note sul programma

Linea 45: stabilisce i parametri del videochip e la zona dove andare a memorizzare gli sprites.

Linee 50-60: altri dati da leggere?

Linea 90: funzione che gestisce il cursore sullo schermo.

```
 1590 END
 3000 PRINT"■DISEGNATORE DI SPRITE MUL
      TICOLORI."
 3002 PRINT"■■PUOI USARE 4 COLORI NEL
      SEGUENTE MODO:"
 3004 PRINT"■■SPRITE MULTICOLORE 1"
 3006 PRINT"SPRITE MULTICOLORE 2"
 3008 PRINT"COLORE SPRITE ORDINARIO"
 3010 PRINT"COLORE DELLO SCHERMO"
 3012 PRINT"■■QUESTO LIMITA LA NOSTRA
      RISOLUZIONE"
 3013 PRINT"DEGLI SPRITES DA 24 A 12 C
      ARATTERI."
 3014 PRINT"■■TUTTAVIA, CON UN PO' D'I
      MMAGINAZIONE,"
 3016 PRINT"SI OTTENGONO RISULTATI SOR
      PRENDENTI■"
 3018 GOSUB4000
 3019 POKE53280,1:POKE53281,1
 3020 PRINT"■■SCEGLI I COLORI FRA I 16
      DISPONIBILI:■"
 3023 PRINTTAB(30)CHR$(18)"■    ■■
      0"
 3024 PRINTTAB(30)CHR$(18)"■    ■■
      1"
 3025 PRINTTAB(30)CHR$(18)"■    ■■
      2"
 3026 PRINTTAB(30)CHR$(18)"■    ■■
      3"
 3027 PRINTTAB(30)CHR$(18)"■    ■■
      4"
 3028 PRINTTAB(30)CHR$(18)"■    ■■ -
      5"
 3029 PRINTTAB(30)CHR$(18)"■    ■■ -
      6"
 3030 PRINTTAB(30)CHR$(18)"■    ■■ -
      7"
 3031 PRINTTAB(30)CHR$(18)"■    ■■ -
      8"
 3032 PRINTTAB(30)CHR$(18)"■    ■■ -
      9"
 3033 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      0"
 3034 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      1"
 3035 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      2"
 3036 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      3"
 3037 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      4"
 3038 PRINTTAB(30)CHR$(18)"■    ■■ - 1
      5"
 3040 INPUT"■■SPRITE MULTICOLORE ZERO"
      ;A2:IFA2<0ORA2>15THENPRINT"■■":G
      OTO3040
 3042 INPUT"■SPRITE MULTICOLORE UNO";A
      3:IFA3<0ORA3>15THENPRINT"■■":GOT
      O3042
 3044 INPUT"■COLORE SPRITE ORDINARIO";
      A4:IFA4<0ORA4>15THENPRINT"■■":GO
      TO3044
 3045 CS=A4
 3046 GOSUB4000
 3048 POKEV+37,A2:POKEV+38,A3:POKEV+39
      ,A4
 3049 POKE53280,0:POKE53281,0
 3050 PRINT"■■":PRINT"■MUOVI IL CURSOR
      E PER LO SCHERMO CON I"
 3051 PRINT"TASTI CURSORE. PER LO STES
      SO SPRITE USA"
 3052 PRINT"L'OPZIONE S DEL MENU'."
 3053 PRINT"■QUANDO SEI SODDISFATTO ME
      TTI I CODICI"
 3054 PRINT"NEI DATA CON L'OPZIONE B.
      SARANNO ALLE"
 3055 PRINT"LINEE 30011-30020."
 3056 PRINT"■■PREMI HOME PER RIPORTARE
       IL CURSORE"
 3057 PRINT"IN ALTO A SINISTRA E CLR/H
      OME PER CAN-  CELLARE LO SPRITE
      COMPLETAMENTE."
 3058 PRINT"■QUANDO CAMBI COLORE PREMI
       1 PER AVERE ILMULTICOLORE 0, 2
      PER IL MULTICO";
 3059 PRINT"LORE 1, E3 PER IL COLORE D
      ELLO SPRITE  ORDINARIO."
 3065 GOSUB4000
 3070 RETURN
 4000 PRINT"■■PREMI 'SPAZIO' PER CONTI
      NUARE"
 4002 GETSP$:IFSP$<>" "THEN4002
 4004 RETURN
29997 REM ----------------------------
29998 REM MEMORIZZAZIONE SPRITE DATA
29999 REM ----------------------------
30011 DATA 13 ,1, 3 , 2 , 4
30012 DATA0,2,128,0,2,128,0
30013 DATA2,128,0,2,128,0,2
30014 DATA128,10,170,128,8,0,0
30015 DATA8,0,0,12,0,0,12
30016 DATA68,0,29,1,0,79,16
30017 DATA64,31,192,16,7,255,244
30018 DATA1,255,253,0,127,244,0
30019 DATA31,208,0,7,64,0,1
30020 DATA0,0,0,0,0,0,0
30021 DATA -1
```

Linee 110-165: stabiliscono i vari parametri degli sprites.
Linee 180-183: gestisce la visualizzazione sullo schermo.
Linea 184: stampa le istruzioni sullo schermo.
Linee 330-370: crea un doppio cursore sullo schermo.
Linee 380-570: controlla i dati inseriti.
Linee 580-620: somma dei punti, creazione schermo e sprite.
Linee 625-627: controlla lo stato dello sprite prima di visualizzarlo.
Linee 730-770: punti rimossi, creazione schermo e sprite.
Linee 808-850: legge i dati dello sprite.
Linee 900-990: espande o rimpicciolisce lo sprite tanto orizzontalmente quanto verticalmente.
Linee 1000-1140: stampa le opzioni sullo schermo.
Linee 1150-1200: rimuove lo sprite, pulisce lo schermo ed azzera i precedenti dati.
Linee 1210-1390: lo sprite si muove lungo un percorso stabilito.
Linee 1400-1440: cambia i colori.
Linee 1450-1590: converte i dati in istruzioni DATA.
Linee 3000-4004: benvenuti allo spettacolo.
Linee 30011-30021: DATA per il versatore di birra.

MEMORIZZAZIONE DEL VIDEO

di Alessandro Barattini

Il programma permette di memorizzare sul nastro qualsiasi videata e in seguito di ricaricarla in memoria: vengono memorizzati tutti i caratteri, i colori ed i registri.
Il problema principale della memorizzazione di schermi è che le istruzioni per il registratore modificano lo schermo presente sul video; quindi è necessario trasferire la videata che ci interessa in un'area di memoria che non viene alterata.
Questa operazione viene fatta da una routine in linguaggio macchina, mentra un'altra routine memorizza poi quest'area sul registratore; non vengono così memorizzati insieme con la videata i messaggi normalmente mandati quando si fà un'operazione di caricamento o memorizzazione. Per il caricamento i dati riguardanti la videata vengono caricati in un'area diversa dalla memoria del video e da qui trasferiti all'area originale. Inoltre, per far sì che non compaia il messaggio "READY", il programma cicla su un'istruzione finché non viene premuto un tasto qualsiasi.
Il programma è diviso in due parti: la prima parte (listato 1) carica in memoria il codice macchina delle routine e quando viene dato il RUN si autocancella. La seconda parte (listato 2) può essere aggiunta a qualsiasi programma, per memorizzare le videate che quest'ultimo crea. Naturalmente, per funzionare, la seconda parte ha bisogno che in memoria sia presente il codice macchina creato dalla prima parte. Se modifichiamo la linea 50060, sostituendo l'istruzione STOP con un RETURN, possiamo memorizzare più videate semplicemente facendo: GOSUB 50000.
Quando il programma gira non appare alcun messaggio; questo perché il programma rimane in attesa di una S per la memorizzazione o di una L per il caricamento di videate già memorizzate. (Se non appaiono segnali di nessun genere è per non rovinare le videate che si vogliono memorizzare). Se premete S o L appaiono le solite istruzioni per il registratore.
Nel momento in cui viene premuta la lettera S il video e la memoria colori vengono trasferiti in un'altra zona della memoria; da qui verranno trasferiti al registratore quando si premono i tasti PLAY e RECORD. La mappa della memoria è mostrata in figura 1.
Quando si carica un video dal nastro, il programma opera nella maniera inversa. Dopo queste operazioni, se si vuole continuare, basta premere un tasto qualsiasi ed appare il messaggio "READY", che, nel caso abbiate fatto un caricamento di un video, ve lo rovina. Non preoccupatevi! Per riavere la videata intatta basterà fare: SYS 49395.
Se si devono memorizzare degli sprite, oltre a delle videate, è necessario memorizzare l'area di memoria dove risiedono i dati relativi ad essi. L'operazione di memorizzazione di sprite va fatta prima di memorizzare il video. La routine che effettua questa operazione è la "Save routine" e la si abilita con un RUN 50100. Essa richiede il nome del file, l'indirizzo di partenza dell'area di memoria e l'ultima loca-

```
 DEC    HEX    DESCRIZIONE
50944  C700   Loc. bassa RAM schermo
50945  C701   " alta           "
50946  C702   Cod. colore car. corrente
50947  C703   Colore cursore
50948  C704   Massimo della mem. schermo
50949  C705 )
  I      I  )
  I      I  ) Mem. caratteri schermo
  I      I  )
51792  CB04 )
51793  CB05 )
  I      I  )
  I      I  ) Registri Vicchip
  I      I  )
52018  CB32 )
52019  CB33 )
  I      I  )
  I      I  ) Mem. colore
  I      I  )
53019  CF1B )
```

```
 DEC    HEX    DESCRIZIONE
49152  C000   Loc. partenza byte basso
49153  C001   "      "      "   alto
49154  C002   Ultima loc. byte basso
49155  C003   "     "      "   alto
49156  C004 )
  I      I  )
  I      I  ) Routine Save
```

```
  I      I  )
49196  C02C )
49197  C02D )
  I      I  )
  I      I  ) Routine Load
  I      I  )
49216  C040 )
49217  C041 )
  I      I  )
  I      I  ) Non usati
  I      I  )
49229  C04D )
49230  C04E   Lung. nome file
49231  C04F )
  I      I  )
  I      I  ) Nome file
  I      I  )
49246  C05E )

49248  C060 )
  I      I  )
  I      I  ) Routine Save schermo
  I      I  )
49387  C0EB )

49392  C0F0 )
  I      I  )
  I      I  ) Routine Load schermo
  I      I  )
49512  C168 )
```

zione di memoria dell'area da memorizzare. Per caricare in memoria dati già memorizzati usate la "Load routine" (queste routine possono essere utilizzate per trasferire qualsiasi zona di memoria, non solo i dati per gli sprite; possono avere quindi un utilizzo in altri programmi).

Alcuni problemi possono nascere quando il programma che crea il video occupa le locazioni da 49152 a 49512, oppure da 50944 a 53019, che sono usate dalle routine. In questo caso è necessario rilocare il codice macchina delle routine, non dimenticando di rilocare i byte da 49152 a 49155, che sono usati come puntatori a specifiche zone di memoria. La figura 2 mostra la mappa della memoria del codice macchina delle routine ed i registri usati.

```
   1 rem ****************************
   2 rem *                          *
   3 rem *    codice   macchina     *
   4 rem *                          *
   5 rem ****************************
  10 restore:print"▒":c=0
  20 for i=49156 to 49216:read a:po
     kei,a:t=t+a:next
  30 if t<>7419 then print"dati sba
     gliati alle linee 110 - 140":s
     top
  40 t=0
  50 for i=49248 to 49387: read a:p
     okei,a:t=t+a:next
  60 if t<>19316 then print"dati sb
     agliati alle linee 140 - 220":
     stop
  70 t=0
  80 for i=49392 to 49512: read a:p
     okei,a:t=t+a:next
  90 if t<>17317 then print"dati sb
     agliati alle linee 220 - 280":
     stop
 100 print"codice macchina caricato
      in memoria":new
 110 data 162,1,169,1,160,3,32,186,
     255,173,78,192,162,79,160,192,
     32,189
 120 data 255,173,0,192,133,20,173,
     1,192,133,21,174,2,192,172,3,1
     92,169
 130 data 20,32,216,255,96,162,1,16
     9,1,160,1,32,186,255,169,0,32,
     189
 140 data 255,169,0,32,213,255,96,1
     65,243,141,0,199,165,244,133,4
     ,173,134
 150 data 2,141,2,199,173,135,2,141
     ,3,199,173,136,2,141,4,199,162
     ,0
 160 data 189,0,4,157,5,199,232,208
     ,247,189,0,5,157,5,200,232,208
     ,247
 170 data 189,0,6,157,5,201,232,208
     ,247,189,0,7,157,5,202,232,208
     ,247
 180 data 189,0,208,157,5,203,232,2
     24,46,208,245,162,0,189,0,216,
     157,51
 190 data 203,232,208,247,189,0,217
     ,157,51,204,232,208,247,189,0,
     218,157,51
```

```
 200 data 205,232,208,247,189,0,219
     ,157,51,206,232,224,232,208,24
     5,169,0,141
 210 data 0,192,169,199,141,1,192,1
     69,27,141,2,192,169,4,141,3,19
     2,32
 220 data 4,192,96,32,45,192,173,0,
     199,133,243,173,1,199,133,244,
     173,2
 230 data 199,141,134,2,173,3,199,1
     41,135,2,173,4,199,141,136,2,1
     62,0
 240 data 189,5,199,157,0,4,232,208
     ,247,189,5,200,157,0,5,232,208
     ,247
 250 data 189,5,201,157,0,6,232,208
     ,247,189,5,202,157,0,7,232,208
     ,247
 260 data 189,5,203,157,0,208,232,2
     24,46,208,245,162,0,189,51,203
     ,157,0
 270 data 216,232,208,247,189,51,20
     4,157,0,217,232,208,247,189,51
     ,205,157,0
 280 data 218,232,208,247,189,51,20
     6,157,0,219,232,224,232,208,24
     5,96
```

```
50000 rem ***************************
50001 rem *                         *
50002 rem *      save e load        *
50003 rem *       del video         *
50004 rem *                         *
50005 rem ***************************
50010 get a$: if a$<>"s" and a$<>"l"
       then 50010
50020 if a$="l" then 50040
50030 sys 49248:stop
50040 sys 49392
50050 get a$: if a$="" then 50060
50060 stop
50100 rem ***************************
50101 rem *       routine save      *
50102 rem ***************************
50110 print"▒▒▒memorizzazione▒▒":n$
      =""
50120 input "nome file ▒▒▒";n$
50130 print:l=len(n$):if l>16 then p
      rint"▒nome troppo lungo▒":stop
50140 for i=49231 to 49246:poke i,32
      :next
```

```
50150 poke 49230,l:if l=0 then 50170
50160 for i=1 to l:poke 49230+i,asc(
      mid$(n$,i,1)):next
50170 print
50180 input "indirizzo di partenza
      ▮▮";p:print
50190 poke 49153,int(p/256):poke 491
      52,p-int(p/256)*256
50200 input "ultima locazione ▮▮";u
50210 if p>u then print"▮▮▮indirizzi
       sbagliati▮":stop
50220 poke 49155,int(u/256)
```

```
50230 poke 49154,u-int(u/256)*256
50240 sys49156
50250 return
50300 rem ***************************
50301 rem *      routine load       *
50302 rem ***************************
50310 print"▮▮▮caricamento▮▮▮"
50320 sys49197
50330 print tab(15)"▮caricamento eff
      ettuato▮"
50340 for i=0 to 1500:next
50350 return
```

CONTO CORRENTE

di Edgardo Di Nicola-Carena ed Andrea Ruffini

Questo programma è nato essenzialmente dallo spunto dato da un esercizio di ragioneria: calcolo degli interessi bancari. In realtà si tratta di un programma adatto ad essere usato praticamente e facilmente adattabile alle proprie esigenze.

Dopo la fine del caricamento, viene posta una prima domanda: "vuoi caricare dati precedenti?". La prima volta che si usa un programma o dopo la chiusura dell'anno precedente, bisogna rispondere "N" per passare direttamente alla visualizzazione di tutti gli eventi. In questo caso, inizialmente, la tabella risulta logicamente vuota. Per inserire degli eventi, cioè segnalare dei prelievi o dei versamenti, bisogna premere il tasto Function 5 ("aggiunta"). In tal caso vengono richieste cinque immissioni: la data, che dev'essere una stringa di cinque lettere, di cui le prime due contengono il giorno e le ultime due il mese; il codice, di due lettere, che il programma memorizza soltanto, senza utilizzarlo nell'elaborazione dei dati; la somma, che dev'essere negativa nel caso di un prelievo e positiva nel caso di un versamento; la valuta,che richiede di essere introdotta come la data; infine, la domanda "dell'anno in corso?" si riferisce alla valuta e domanda se la valuta è riferita allo stesso anno dei versamenti od a quello dopo.

Per tutte queste domande esiste un controllo per verificare l'esattezza dell'introduzione. Dopo che si è risposto a tutte le domande, si possono ancora modificare i dati introdotti, premendo la lettera a sinistra della domanda interessata. Si può anche abbandonare del tutto l'operazione scartando i dati introdotti. Infine, si possono introdurre i dati nella lista.

Quando sono stati introdotti molti dati, lo schermo non può contenerli tutti, così è possibile usare "F1" ed "F3" per muoversi rispettivamente in avanti ed indietro.

L'opzione "ricerca" (tasto F7) permette di ricercare un qualsiasi evento, in base alla sua data. Se è possibile il computer visualizza tutti gli eventi che hanno quella data prima di tutti gli altri sullo schermo. Se ciò non è possibile, gli eventi compaiono comunque sullo schermo e vengono trovati immediatamente, anche perché devono essere stati introdotti in ordine di data.

Alla fine dell'anno si potranno calcolare gli interessi usando l'opzione "interessi" (F4). Gli interessi vengono poi aggiunti automaticamente alla lista, con il codice CI. Per conoscere gli interessi che si vanno sviluppando in un qualsiasi momento durante l'anno, bisogna prima assicurarsi di avere almeno una copia dei dati in memoria e poi estinguere il conto. Gli interessi, comunque, non sono riconosciuti dalla banca fino al 31 dicembre.

Per salvare i dati presenti in memoria e ricaricarli la volta successiva in cui sono state fatte operazioni in banca, bisogna premere il tasto F2. In caso di errore, per ripetere l'operazione, dare il comando GOTO 5500.

100-210: inizializzazione
300-350: caricamento dati
500-690: stampa la tabella e l'elenco delle opzioni disponibili
700-760: attende che venga premuto un tasto e si comporta in conseguenza
1000-1040: va avanti nella lista
2000-2030: va indietro nella lista
3000-3900: introduce un nuovo evento
4000-4530: ricerca una data nella lista
5000-5530: memorizza i dati che ci sono in memoria
6000-7120: calcola e stampa gli interessi
8000-8070: routine di controllo della data.

```
0 rem
1 rem programma di conto corrent
  e
2 rem di edgardo di nicola-caren
  a
3 rem                  e
4 rem         andrea ruffini
5 rem
```

```
100 rem inizializzazione
110 dim d$(500),v$(500),s(500),l(5
    00),c$(500),m(12),w$(500),j(50
    0)
115 restore:   forn=1to12:readm(n)
    :next
120 neventi=0:puntev=1
130 data31,29,31,30,31,30,31,31,30
```

```
   ,31,30,31
160 if da$="s" then m(2)=29
200 input"███   vuoi caricare dat
    i precedenti";da$
210 if da$="n"then goto500
300 rem caricamento dati
310 open 1,1,0,"c/c.dati"
320 n=1
330 input# 1,d$(n):input# 1,c$(n):
    input# 1,s(n):input# 1,v$(n):
    input# 1,l(n)
340 if st=0 then n=n+1:goto330
350 neventi=n:close 1
500 rem visualizzazione eventi
510 print"█┌──────┬───┬─┬──────────┬──
    ─┬─┬──────────┐";
520 print"|data  |co|s| somme  |va
    l. |s| saldi   |";
530 print"├──────┼───┼─┼──────────┼──
    ─┼─┼──────────┤";
535 remscrittura eventi
540 for even=puntev to puntev+14
545 if even>neventithen even=punte
    v+14:goto630
550 print"|";left$(d$(even),2);"/"
    ;right$(d$(even),2);"|";c$(eve
    n);"|";
560 ifs(even)>0thenprint"+";
565 ifs(even)=0thenprint" ";
570 ifs(even)<0thenprint"-";
580 printspc(10-len(str$(s(even)))
    );abs(s(even));"████████████|██
    ██████|";
585 printleft$(v$(even),2);"/";ri9
    ht$(v$(even),2);"|";
600 ifl(even)>0thenprint"+";
605 if l(even)=0 then print" ";
610 ifl(even)<0thenprint"-";
620 printspc(10-len(str$(l(even)))
    );abs(l(even));"████████████|██
    ██████|";
630 nexteven
640 print"└──────┴───┴─┴──────────┴──
    ─┴─┴──────────┘"
650 rem scrittura menu
660 print"████████████████████████┘───
    ──────────────────────────────
    ─────I";
670 print"|  f1 => avanti        f3
    => indietro     |";
680 print"|  f5 => aggiunta      f7
    => ricerca      |";
685 print"|  f2 => registraz.    f4
    => interessi    |";
690 print"J───────────────────────────
    ──────────────K";
700 remattende pressione tasto
710 get a$
720 if a$="█"then1000
730 if a$="█"then2000
740 if a$="█"then3000
750 if a$="█"then4000
752 if a$="█"then goto 5000
754 if a$="█"then goto 6000
760 goto710
1000 rem avanti nella lista
1010 puntev=puntev+10
1020 if puntev>neventi-14thenpuntev
     =neventi-14
1030 ifpuntev<1thenpuntev=1
1040 goto500
2000 rem indietro nella lista
2010 puntev=puntev-10
2020 ifpuntev<1thenpuntev=1
2030 goto500
3000 rem aggiunta di un evento
3005 ifneventi=500thengoto500
3007 neventi=neventi+1:pu=neventi-1
     4:if pu<1then pu=1
3010 printtab(4);"███    introduzion
     e di un nuovo evento"
3020 print"█─────────────────────────
     ──────────────────"
3030 print"  a) data (gg/mm)"
3040 print"███ b) codice (due lette
     re)"
3050 print"███  c) somma"
3060 print"███  d) valuta (gg/mm)"
3067 print"███  e) dell'anno in cors
     o (s/n)"
3068 print"█─────────────────────────
     ──────────────────"
3070 gosub3570:gosub3595
3080 gosub3700:gosub3750
3090 gosub3800
3100 print"███    premi a-e per corr
     eggere un dato"
3110 print"█ premi r per rinunciare
      all'operazione"
3120 print"█     oppure 'return' p
     er tabellarlo";
3130 get a$
3135 v$(neventi)=left$(dv$,2)+dt$+r
     ight$(dv$,2)
3140 if a$="a"then gosub 3570
3150 if a$="b"then gosub3595
3160 if a$="c"then gosub3700
3170 if a$="d"thengosub3750
3175 if a$="e"then gosub3800
3180 if a$="r"then neventi=neventi-
     1:goto500
3190 if a$=chr$(13)then l(neventi)=
     l(neventi-1)+s(neventi):goto50
     0
3200 goto 3130
3570 rem introduzione data
```

```
3580 input"";da$
3585 if len (da$)<>5then 3580
3590 gosub8000:if f then 3580
3592 d$(neventi)=left$(da$,2)+right$(da$,2):return
3595 rem introduzione codice
3600 input"";cod$
3601 iflen(cod$)<>2then goto3600
3603 c$(neventi)=cod$
3605 return
3700 rem introduzione somma
3710 input "";s(neventi)
3720 if s(neventi)<>int(s(neventi))orabs(s(neventi))>=1e9then3700
3730 return
3750 rem introduzione valuta
3760 input "";da$
3765 iflen(da$)<>5then3760
3770 gosub8000:if f then 3760
3780 dv$=da$
3790 return
3800 rem introduzione anno valuta
3805 input"";da$
3810 if da$="s" then dt$="a":return
3820 if da$="n" then dt$="b":return
3830 goto3805
3900 return
4000 rem ricerca di una data
4010 print"             ricerca di un evento"
4020 print"──────────────────────────────────────"
4030 print"  data dell'evento (gg/mm)"
4040 input "";da$
4041 if len (da$)<>5then goto4040
4042 gosub 8000:if f then goto 4040
4048 ri$=left$(da$,2)+right$(da$,2)
4100 rem ricerca nella lista
4110 n=neventi
4120 if d$(n) = ri$ thenn=n-1:goto 4150
4130 n=n-1:if n=0 then goto 4500
4140 goto4120
4150 if d$(n) = ri$ then n=n-1:goto 4150
4160 n=n+1
4400 puntev=n:if puntev>neventi-14then puntev=neventi-14
4405 if puntev<1then puntev=1
4410 goto500
4500 print" non esistono eventi svolti nella data"
4510 print"         che hai indicato!"
4520 for n=1 to 2000:next
4530 goto 500
5000 rem registrazione dati su cassetta
5010 print"            registrazione dati"
5020 print"──────────────────────────────────────"
5030 print"  prepara il registratore per la"
5040 print"              registrazione"
5045 print" ";
5050 rem registrazione dati
5060 open 1,1,1,"c/c.dati"
5070 for n=1 to neventi
5080 print# 1,d$(n):print# 1,c$(n):print# 1,s(n):print# 1,v$(n):print# 1,l(n)
5090 next: close 1
5100 input"         vuoi la verifica";da$
5110 if da$="n"then goto 500
5120 open 1,1,0,"c/c.dati"
5130 n=1
5140 input#1,da$:if da$<>d$(n)then goto 5500
5150 input#1,da$:if da$<>c$(n)then goto 5500
5160 input#1,dat:if dat<>s(n)then goto 5500
5170 input#1,da$:ifda$<>v$(n)thengoto5500
5180 input# 1,dat:if dat<>l(n)thengoto5500
5190 if st=0thenn=n+1:goto5140
5200 close 1
5210 goto 500
5500 rem errore!
5505 close 1
5510 print"      si e' verificato un errore."
5520 print"              ripeti l'operazione"
5525 for n=1to3000:next
5530 goto5000
6000 rem calcolo interessi
6010 print"            calcolo interessi"
6020 print"──────────────────────────────────────"
6030 print"   attendi qualche istante, per favore"
```

```
6040 rem trasferimento matrici
6050 for n=1 to neventi
6060 w$(n)=v$(n):j(n)=s(n)
6070 next
6075 rem riordinamento
6080 for n2=neventi to 2 step -1
6090 for n= 1 to n2-1
6100 gio=val(left$(w$(n),2)):mese=v
     al(right$(w$(n),2))
6110 g2=val(left$(w$(n+1),2)):m2=va
     l(right$(w$(n+1),2))
6115 if mid$(w$(n),3,1)>mid$(w$(n+1
     ),3,1)then 6200
6117 if mid$(w$(n),3,1)<mid$(w$(n+1
     ),3,1)then 6300
6120 if mese>m2 then 6200
6130 if mese<m2 then 6300
6140 if gio>g2 then 6200
6150 goto 6300
6200 da$=w$(n):w$(n)=w$(n+1):w$(n+1
     )=da$
6210 dat=j(n):j(n)=j(n+1):j(n+1)=da
     t
6300 next:next n2
6310 rem stampa della tabella
6320 print"▣ ┌─────────┬──┬────────────────┬──
     ─┬──┬────────────────┐"
6330 print" |val. |s|   somme   |gg
      |s|   n/100    |"
6340 print" ├─────────┼──┼────────────────┼──
     ┼──┼────────────────┤"
6350 som=0:snum=0:for n=1 to nevent
     i-1
6400 gio =val(left$(w$(n),2)):mese=
     val(right$(w$(n),2))
6410 g2=val(left$(w$(n+1),2)):m2=va
     l(right$(w$(n+1),2))
6420 gg=0
6430 mm=mese
6435 ifmm=m2then goto6450
6440 gg=gg+m(mm):mm=mm+1:if mm=13 t
     hen mm=1
6445 goto6435
6450 gg=gg+g2-gio
6455 som=som+j(n)
6460 gosub 7000
6570 nextn
6580 gio=val(left$(w$(n),2)):mese=v
     al(right$(w$(n),2))
6590 m2=12:g2=31
6600 gg=0
6610 mm=mese
6615 if mm=m2then 6630
6620 gg=gg+m(mm):mm=mm+1:if mm=13 t
     hen mm=1
6625 goto6615
6630 gg=gg+g2-gio
6635 if mid$(w$(ne),3,1)="b" then g
     g=gg-366
6645 som=som+j(neventi)
6650 gosub 7000
6670 print" └─────────┴──┴────────────────┴──
     ┴──┴────────────────┤"
6680 printtab(24);"|";
6690 if snum<0thenprint"-";
6700 if snum=0thenprint" ";
6710 if snum>0then print"+";
6720 print"|";spc(11-len(str$(int(a
     bs(snum)))));int(abs(snum));"▌
     |"
6730 printtab(24);"└──┴────────────────┘"
6735 printtab(25);"───────────────────"
6740 input"▣▣   tasso d'interesse";
     r
6745 ci=int(r*snum/360)
6750 neventi=neventi+1:d$(neventi)=
     "31a12":c$(neventi)="ci":s(nev
     enti)=ci
6760 v$(neventi)="31a12":l(neventi)
     =l(neventi-1)+ci
6765 pu=ne-14:if pu<1thenpu=1
6770 goto 500
7000 rem stampa interessi
7020 num=som*gg/100:snum=snum+num
7030 print" |";left$(w$(n),2);"/";r
     ight$(w$(n),2);"|";
7040 if som<0thenprint"-";
7050 if som=0thenprint" ";
7060 if som>0then print"+";
7070 print"|";spc(10-len(str$(som))
     );abs(som);"▌|";spc(4-len(str$
     (gg)));
7075 printright$(str$(gg),len(str$(
     gg))-1);"|";
7080 if num<0thenprint"-";
7090 if num=0 then print" ";
7100 if num>0thenprint"+";
7110 print"|";spc(11-len(str$(int(n
     um))));abs(int(num));"▌|"
7120 return
8000 rem controllo data (da$)
8010 ifleft$(da$,1)<"0"orleft$(da$,
     1)>"9"then f=1:return
8020 ifmid$(da$,2,1)<"0"or mid$(da$
     ,2,1)>"9"then f=1:return
8030 ifmid$(da$,4,1)<"0"ormid$(da$,
     4,1)>"9"thenf=1:return
8040 ifright$(da$,1)<"0"orright$(da
     $,1)>"9"thenf=1:return
8050 mese=val(right$(da$,2)):ifmese
     =0ormese>12thenf=1:return
8060 gio=val(left$(da$,2)):if gio>m
     (mese)or gio=0thenf=1:return
8070 f=0:return
```

L'ELETTRONICA IN LABORATORIO

GUIDA INDISPENSABILE PER CHI VIVE LA VITA DI LAVORO A CONTATTO CON L'ELETTRONICA

a cura di E. CHIESA